



Du Langage Naturel à la Spécification – Application à la Spécification de Services de Télécommunication

Alain-Jérôme Fougères

► To cite this version:

Alain-Jérôme Fougères. Du Langage Naturel à la Spécification – Application à la Spécification de Services de Télécommunication. SETIT'04, Mar 2004, Sousse, Tunisie. pp.1-8. hal-00575209

HAL Id: hal-00575209

<https://hal.science/hal-00575209>

Submitted on 9 Mar 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Du Langage Naturel à la Spécification – Application à la Spécification de Services de Télécommunication

Alain-Jérôme Fougères

¹Laboratoire M3M - UTBM

Rue du Château Sévenans

90010 BELFORT - France

alain-jerome.fougeres@utbm.fr

Résumé— La rédaction de spécifications techniques, écrites la plupart du temps en langage naturel, est devenue une activité considérable. Parallèlement la nécessité de réduire les temps de développement de systèmes complexes, comme ceux des télécommunications, est une priorité. Une condition pour atteindre cet objectif consiste à formaliser le maximum de spécifications produites. Nous essayerons donc de montrer la possibilité d'une certaine automatisation du passage de l'informel au formel, grâce à des méthodes et outils fiables, susceptibles d'assister un expert humain en spécification. Nous proposons pour cela un processus de formalisation qui s'appuie sur une représentation intermédiaire des spécifications avec le formalisme des graphes conceptuels, avant de dériver une description formelle en Z. Puis, dans une deuxième approche, nous exposerons les éléments méthodologiques liés à un processus de conception orientée agent, cette fois semi-formel. Pour illustrer notre présentation et lui conférer une meilleure lisibilité, nous avons choisi un exemple simple de spécification que nous appliquerons à chacune des phases du processus de formalisation.

Mots Clés— Traitement du langage naturel, système multi-agents, spécifications formelles, langage Z, spécifications objets, A_UML.

1 PRESENTATION

— Dans les premières phases du développement de systèmes complexes, comme ceux des télécommunications, les spécifications de logiciels, de services ou de matériels, sont des documents volumineux le plus souvent rédigés en langage naturel. De ce fait, l'activité induite par la rédaction de ces spécifications devient considérable : une multitude de pages sont produites tous les jours. De ce fait, est-il possible d'exploiter des procédures facilitant le développement et l'écriture des spécifications ? et peut-on bénéficier d'une aide pour l'archivage de ces masses de données et pour le maintien de cohérence durant leur développement et leur exploitation ?

Les spécifications contribuent à la qualité et à la fiabilité des services développés. De plus, elles constituent le document de référence – consulté et exploité – de la phase de développement du service jusqu'à sa maintenance, en passant par la validation. La maîtrise de l'étape de spécification fonctionnelle devient alors primordiale, puisqu'elle conditionne la réalisation du service. Toute démarche facilitant la rédaction de spécifications, avec le respect des qualités

attendues, contribue à la réduction des temps inhérents à la réalisation de cette étape, comme des étapes suivantes. Tenant compte de ces remarques, nous essayerons de montrer la possibilité d'une certaine automatisation du passage de l'informel au formel, grâce à des méthodes et outils fiables, susceptibles d'assister un expert humain en spécification.

Nous nous sommes fixés comme objectif de définir un système informatique d'aide à l'exploitation rationnelle de spécifications de systèmes en langage naturel, utilisant les ressources d'un formalisme de représentation des connaissances pour modéliser le contenu informationnel des spécifications. Nous proposerons pour cela un processus de formalisation de spécifications rédigées en langage naturel qui s'appuie sur une représentation intermédiaire des spécifications dans le formalisme des *graphes conceptuels* [14], pour produire une description formelle dans un langage de spécification cible : le *langage Z* [16].

Après avoir étudié le contexte de la spécification formelle, nous nous sommes concentrés sur une

approche orientée agent, adapté aux systèmes coopératifs et distribués. Les principales caractéristiques des agents [2] (autonomie, adaptabilité, coopération et communication) permettent, d'une part, de gérer efficacement des composants distribués, hétérogènes et autonomes, et, d'autre part, de faciliter les échanges d'informations et le partage de ressources entre les composants (interaction, communication et coopération). Les agents peuvent être de type *application*, *coordinateur/médiateur*, *système* et *interface*. Le système d'agents doit, quant à lui, assurer l'organisation et le contrôle de la communauté d'agents. A partir de la description fournie dans la spécification en langage naturel, nous avons défini un nouveau processus d'assistance pour développer la conception agent de systèmes en UML.

Cet article est structuré en deux parties consacrées à nos deux approches d'aide à la spécification. La première partie présente successivement la description du domaine de l'assistance à la spécification formelle, la présentation de notre processus de formalisation de spécifications à partir des spécifications rédigées en langage naturel. La seconde partie décrit le contexte des systèmes multi-agents, puis développe les étapes de construction d'une description semi-formelle en UML de spécifications rédigées en langage naturel. L'illustration de ces deux approches sera réalisée à l'aide d'un même simple de spécification simple d'une transmission de messages. Nous terminons l'exposé de nos études par une discussion sur la faisabilité et les limites de ces démarches.

2 L'AIDE A LA SPECIFICATION FORMELLE

Malgré l'ensemble conséquent d'outils d'aide à la spécification et de langages plus ou moins formels, l'essentiel des spécifications est toujours écrit en langage naturel et le restera certainement longtemps encore. Pour limiter au maximum l'aspect informel des spécifications, les experts définissent souvent le cadre général de l'application à l'aide de langages de modélisation ou de méthodologies (RdP, SADT, UML) et n'utilisent qu'un langage naturel contraint pour les descriptions de plus bas niveau. Aussi, pour fournir une aide significative à la rédaction de spécifications formelles nous avons envisagé deux axes : l'assistance technique au développement de spécifications et la prise en compte des aspects sémantiques intervenant dans le processus de formalisation.

2.1 Objectif

L'objet de cette recherche est l'étude de méthodes, techniques et outils permettant d'offrir une aide aux rédacteurs de spécifications dans leur activité de formalisation. Un résumé de notre problématique peut se formuler au moyen de l'interrogation suivante :

comment participer au processus de formalisation

(i.e. passage de l'informel au formel), à partir de spécifications écrites en langage naturel ?

2.2 Les Hypothèses

Avant de proposer une solution à la question précédente, nous dressons ci-dessous la liste des principales hypothèses qui ont guidé notre approche et nous ont ainsi permis d'élaborer l'architecture d'un système de formalisation de spécifications [4].

— **1^{ère} hypothèse** : *il n'est pas envisageable de traduire directement une spécification informelle en une spécification formelle.*

Cette hypothèse est directement corrélée à la problématique de traduction d'un langage naturel en un langage formel. De plus, l'observation de l'activité de spécification corrobore cette hypothèse, puisqu'il est trivial d'énoncer qu'une spécification formelle s'élabore progressivement (i.e. par raffinement).

— **2^{ème} hypothèse** : *il est pertinent d'utiliser une représentation intermédiaire pour passer de l'informel au formel.*

Nous venons de citer le paradigme de raffinement pour la construction de spécifications formelles, et aurions pu nous en inspirer, mais celui-ci n'est pas suffisamment défini, pour établir de façon univoque le nombre de passes de raffinement nécessaires pour une spécification donnée. Aussi, nous sommes convaincus que le passage par une représentation intermédiaire, conçue comme un pivot pour la formalisation, répond à notre problématique. Notre choix pour la représentation des connaissances s'est orienté vers le formalisme des graphes conceptuels (GC), considérant qu'il respecte la définition des critères indispensables à notre objectif. Le passage par cette représentation intermédiaire ne lève pas pour autant toutes les difficultés, il permet cependant de mieux les caractériser – la complexité de la traduction d'une sémantique naturelle en une sémantique formelle demeure, mais elle se dédouble et devient plus facile à maîtriser.

— **3^{ème} hypothèse** : *il est nécessaire de construire une description logique issue de la représentation intermédiaire, qui puisse servir de pivot pour la traduction vers différents langages de spécification formels.*

Cette hypothèse est le résultat d'une constatation unanime chez les rédacteurs de spécifications : il est rare qu'un langage de spécification formel se prête correctement à la modélisation de la totalité d'une spécification – la spécificité de chacun d'eux les rend plus aptes à modéliser un aspect plus qu'un autre. Il n'est pas interdit de penser qu'un langage futur aura les propriétés de généralité, mais tel n'est pas encore le cas. Une des conséquences de ceci est l'apparition du paradigme de multi-spécification [13], consistant à utiliser, dans une même spécification, le langage le plus propice à fournir une description pour un module donné. Avec la construction d'une description logique nous respectons le paradigme de multi-spécification en offrant la possibilité de traduire son contenu dans le

langage formel cible – moyennant la disponibilité du traducteur adéquat.

— **4^{ème} hypothèse** : les éléments de la représentation intermédiaire sont trop près du LN (sémantique littérale), il est souvent nécessaire de les transformer au moyen de règles pour les rendre pertinents au niveau de la spécification formelle.

Cette quatrième hypothèse est apparue après une première série d'expérimentations, dans lesquelles nous construisions une spécification formelle issue de la représentation sous forme de graphes conceptuels qui contenaient la sémantique littérale des textes de spécifications. Cette formalisation était assez éloignée de celle produite par les rédacteurs de spécifications. La raison principale de cette distance tenait dans la trop forte exploitation de relations thématiques (de type Agent, Objet, etc.) propres à représenter le sens des phrases mais nullement évocatrices pour des spécifications techniques.

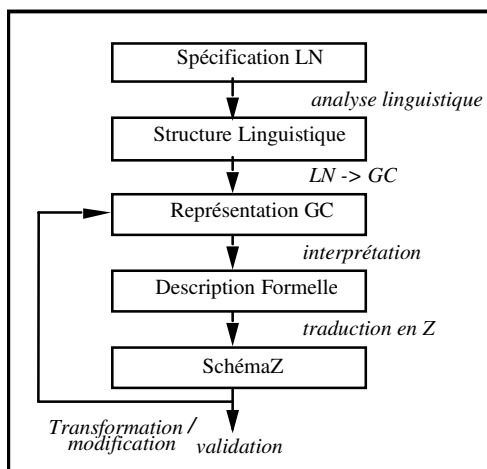


FIG. 1 – Chaîne de traitements des spécifications : acquisition des connaissances, analyse linguistique, interprétation et obtention d'un modèle Z.

2.3 Le Processus de Formalisation

Pour répondre à la démarche de formalisation, nous proposons un enchaînement de traitements sur les spécifications initiales, susceptibles de fournir une description formelle. Le séquençement des différentes étapes de ce processus (Fig. 1), faisant suite à une étape préalable d'acquisition de connaissances, représente la mise en œuvre de notre objectif, pondéré par l'ensemble des hypothèses énoncées ci-dessus.

3 ILLUSTRATION DE CETTE APPROCHE

Nous avons expérimenté chacune des phases de la figure 1 sur une spécification réelle, issue des NEF¹.

¹Normes d'Exploitation et de Fonctionnement des autocommutateurs (France Télécom), ensemble de documents volumineux de spécifications écrites en LN.

Cependant, pour la clarté de l'énoncé, nous présentons dans la suite les résultats obtenus sur une spécification plus élémentaire et de taille réduite (Fig. 2).

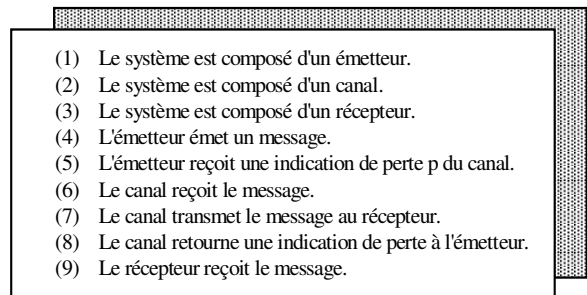


FIG. 2 – Énoncé simplifié d'une spécification de "transmission de messages"

Le compte rendu de ces expérimentations s'articulera autour de trois sections décrivant successivement la phase préliminaire d'acquisition des connaissances, les différents niveaux de représentations des spécifications (linguistique, sémantique et formelle) et une évaluation de notre démarche.

3.1 Acquisition des Connaissances

Cette étape préalable consiste en l'extraction d'informations lexicales contenues dans le texte, afin de déterminer les liens privilégiés que les mots ont entre eux. Une simple étude de co-occurrence de mots, basée sur une analyse de proximité lexicale, permet de révéler la présence de mots composés, de locutions, de relations prédictives et de schémas de phrases propres au domaine. L'utilisation conjointe à cette analyse fréquentielle de techniques de filtrage statistique, telle que l'*information mutuelle*, permet d'affiner et d'augmenter la pertinence des résultats obtenus.

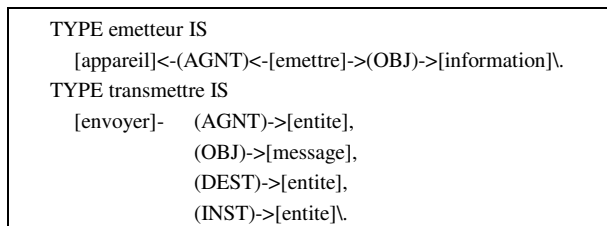


FIG. 3 – Conceptualisation à partir du dictionnaire

La deuxième phase d'acquisition des connaissances, consiste à extraire, du dictionnaire, des définitions de termes retenus comme concepts, afin de les décrire dans un dictionnaire sémantique sous la forme de graphes conceptuels. Le but de ce traitement est d'accroître les connaissances de la base en y intégrant des *informations sémantiques lexicales* issues d'un dictionnaire et ainsi d'élargir l'ontologie de notre modèle grâce à l'apport de définitions pour chaque concept. Une fois analysé le contenu d'une définition, il

est alors possible de construire le graphe conceptuel correspondant et de l'inclure dans une *base canonique*.

3.2 Représentations Linguistiques

Il est habituel en linguistique de considérer cinq niveaux de la langue écrite : morphologie, lexique, syntaxe, sémantique, pragmatique. Les procédures de traitement des langues naturelles reposent sur un corpus de connaissances dans chacun de ces cinq niveaux linguistiques. On peut certes discuter de l'interdépendance de ces différents niveaux, mais nous considérons que cette stratification a le mérite d'être claire à exposer et qu'elle permet de présenter sans complexité tous les détails de nature linguistique.

3.2.1. Le niveau morpho-lexicale

Il s'agit lors de cette phase de séquencer les phrases analysées afin d'obtenir une suite de *lexèmes*, après identification des mots simples, des mots composés et des expressions figées. Cette analyse est rendue possible par l'utilisation d'un lexique apparenté au DELAS² et d'un module d'analyse flexionnelle pour la description des formes fléchies (conjugaison, genre, nombre). En plus de la reconnaissance des mots, cette étape permet leur distribution en classes (catégories syntaxiques), telles que *nom*, *verbe*, *adjectif*.

3.2.2. Le niveau syntaxique

Les stratégies d'analyse syntaxique sont très nombreuses : analyses en constituants, grammaires formelles, étiquetage grammatical probabiliste, etc. Notre propos, n'est pas ici de faire un panorama de ces méthodes, mais plutôt d'exposer succinctement le formalisme LFG [10] que nous avons choisi.

Ce formalisme se décompose en trois niveaux :

- la *c-structure* (analyse en constituant), décrite au moyen de règles de productions d'une grammaire hors-contexte, représente les structures syntaxiques ;

- la *f-structure* (description fonctionnelle), composée de paires fonction-valeur, met en évidence les fonctions grammaticales telles que *Sujet*, *Objet*, etc. Une fonction particulière, nommée *Pred*, met en correspondance les fonctions syntaxiques et les rôles sémantiques d'un prédicat ;

- la *s-structure* (structure sémantique), projection sémantique à partir de la *c-structure* qui permet de ne retenir que la structure prédicative.

Les grammaires LFG ajoutent donc à la construction de la structure syntaxique la formation d'une structure fonctionnelle de la phrase (les liens entre les groupes : analyse logique), spécifiée au moyen de schémas associés aux règles de grammaire. La validité de ces structures est régie par des principes généraux de bonne formation, qui permettent de rejeter certaines analyses

syntactiques incorrectes. La deuxième caractéristique du formalisme LFG est son aspect lexical. Il donne en effet une place importante au lexique, en effectuant directement sur les interprétations des mots toutes les transformations pertinentes (passivation, forme pronominales de certains verbes, etc.).

3.2.3. L'analyse sémantique

L'analyse sémantique a pour rôle d'identifier les différents acteurs, objets, instruments, lieux et modalités intervenant dans une phrase. Mais produire une bonne représentation de la phrase suppose aussi d'être capable de traiter un certain nombre de phénomènes présents dans toutes les langues : les ambiguïtés, les paraphrases, les références, ...

Le formalisme de représentation des connaissances sémantiques étant fixé, cette analyse consiste alors en la traduction sémantique de la structure syntaxique sous forme de graphes conceptuels (Fig. 4). Pour cela, nous nous inspirons des grammaires de cas [3] qui déterminent les différents rôles thématiques remplis par les constituants d'une phrase à l'aide des informations acquises sur l'ordre des mots, les prépositions, les verbes et le contexte. En d'autres termes, l'analyseur détermine la façon dont les syntagmes nominaux d'une phrase sont reliés aux verbes – le rôle sémantique spécifiant comment un objet participe à la description d'une action. Les sources d'informations activées pour les besoins de cette analyse sont : le lexique sémantique et la description fonctionnelle LFG.

- Étape 1** : rechercher, dans le lexique sémantique, le graphe *G* du verbe.
- Étape 2** : construire le graphe *G'* pour le sujet et essayer de le joindre avec les relations *AGNT*, *INST*, *OBJ* selon *G*.
- Étape 3** : pour chaque argument du verbe, joindre le graphe correspondant au graphe *G*, avec la relation définie par la proposition qui introduit l'argument.
- Étape 4** : attacher les graphes correspondant aux modificateurs (préposition, pronom relatif, etc.) au graphe *G*.
- Étape 5** : contextualiser le graphe entier (ex : négatif, si le verbe est lui-même négatif).
- Étape 6** : joindre les graphes correspondant aux clauses subordonnées.

FIG. 4 – GC à partir de la structure fonctionnelle

3.3 Une Représentation des Connaissances

Le problème fondamental de la représentation informatique des connaissances est l'élaboration d'une notation suffisamment précise et formelle (cadre représentationnel) permettant cette représentation. L'utilisation de ce cadre consiste alors à spécifier une *base de connaissances* contenant un ensemble de *faits*

²Dictionnaire du LADL contenant environ 80000 mots simples avec leurs références catégorielles [7].

au format établi. Le monde décrit par un *cadre représentationnel* est une collection d'individus et de relations entre ces individus qui spécifient un état susceptible de subir des transformations. Trois familles de cadres représentationnels peuvent être évoquées suivant les critères de leur construction :

- à partir d'assertions vraies sur les états, il s'agit de cadres logiques ;
- à partir de la représentation des individus et de leurs relations, le cadre est du type *réseau sémantique* ;
- à partir de la description de changements d'états, les cadres sont dits procéduraux.

Dans le modèle des graphes conceptuels les objets élémentaires sont les *concepts* et les *relations* (individus et relations), ce qui apparente le formalisme à la famille des réseaux sémantiques. Chaque proposition (fait) est représentée par un graphe conceptuel construit au moyen d'arcs dirigés reliant concepts et relations. Des règles offrant la possibilité de joindre ou de dissocier des graphes conceptuels (joints et projections) sont données. Une correspondance formelle (isomorphisme) avec la logique des prédicats du premier ordre est établie pour un noyau de base.

Cette description sommaire des GC met en évidence la mixité des qualités de ce formalisme de représentation des connaissances : un modèle assez "psychologique" dans sa forme, ce qui lui confère une grande lisibilité, et un système de preuves mathématiques ayant des fondements axiomatiques solides, ce qui le rend formel.

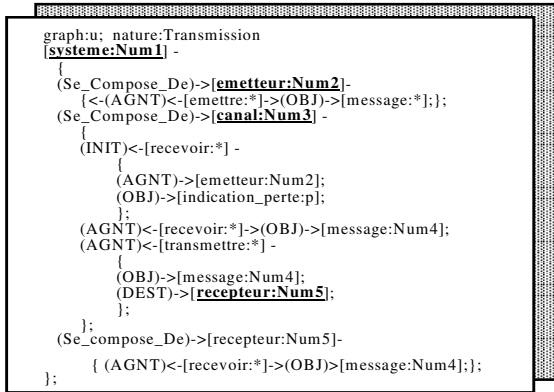


FIG. 5 – Graphe résultant de la jointure de u1 à u9 ; nous avons noirci les concepts liés aux jointures

3.4 La Représentation en Langage Z

Une spécification en Z est constituée par une séquence de paragraphes comportant des *schémas*, des *variables* et des *types de base*. À toute expression apparaissant dans une spécification en Z est associé un type unique (ensemble, produit cartésien, schéma). Un schéma est constitué d'une *signature* (collection de variables typées) et d'une *propriété* sur cette signature, appelée partie prédictive. Pour établir la description

formelle correspondante à une spécification informelle, nous avons procédé par analogie avec le mécanisme de construction déployé par un expert humain (Fig. 6). On commence par extraire les éléments de la description formelle, on identifie et on insère les éléments indispensables qui ne sont pas compris dans le module à spécifier, puis on établit les formules logiques (pré-conditions, post-conditions, propriétés) correspondant aux éléments collectés et à leurs liens. La phase finale est constituée par la modélisation en Z des GC construits à partir de la spécification informelle.

init : 1 GC par phrase analysée.

Étape 1 : effectuer la jointure des graphes d'une section : $\text{Joint}(u_1, \dots, u_n) \rightarrow u$.

Étape 2 : recenser les références externes à la section.

Étape 3 : recenser les paramètres, variables et relation de la description :

- . liste des référents individuels (instance d'objet),
- . liste des concepts en fonction du type de référent,
- . liste des relations en fonction de l'arité.

Étape 4 : exécuter l'algorithme GC \rightarrow Z qui fournit le Schéma U.

FIG. 6 – Construction automatique de description formelle à partir des GC représentant une spécification

Pour illustrer cette méthodologie, nous l'appliquons sur les phrases (1, 4 et 7) de la figure 4 avec les graphes correspondants (*u1*, *u4*, *u7*) ce qui nous permet de dériver le schéma Z correspondant à *u* (Fig. 7).

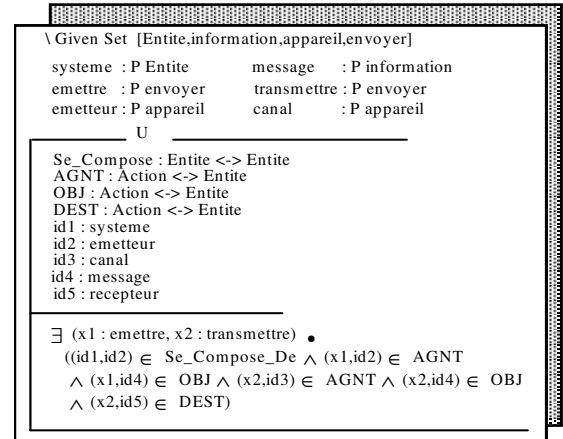


FIG. 7 – Schéma Z représentant le graphe *u*

Les éléments algorithmiques de la traduction d'un graphe conceptuel en Z étant présentés dans [4], on retiendra essentiellement les deux traitements opérés sur les concepts et les relations. Les concepts {*canal*, *emetteur*,...} sont traduits dans le modèle Z suivant leur référent respectif (individuel, générique ou variable) leur super-type dans le treillis des types de concepts permet d'établir une relation d'appartenance à un

ensemble, conformément à la sémantique du langage Z. Les relations {*AGNT*, *OBJ*,...} sont traduites dans les parties déclarative et prédicative, en termes de définitions typées et relations d'appartenances.

4 DES AGENTS POUR LA COOPERATION

Le principal intérêt des SMA est qu'ils permettent de distribuer des agents, entités communicantes, autonomes, réactives et dotées de compétences [2]. Pour réaliser un SMA selon ces critères, il faut doter chaque agent des trois propriétés suivantes : indépendance, communication et intelligence (expertise, savoirs-faire). Il nous faut aussi définir l'architecture des agents (fonctions et interactions cognitives), ainsi que la structuration des connaissances nécessaires pour leurs différentes activités.

4.1 Eléments de Modélisation

La définition de nos agents est adaptée du modèle à 3 niveaux de l'opérateur de Rasmussen (comportement réflexe, comportement à base de règles, comportement à base de connaissances avec interprétation, décision et plan). Nous l'avons interprété comme modèle de processus de nos agents dont les comportements sont adaptés aux tâches qu'ils réalisent :

— **Def Agent** ::= <Communication/langage, Perception, Buts/intentions, Décision/plan, Mémoire, Identification/Interprétation, Actions/Réactions>.

4.2 Architecture d'un Agent Cognitif

L'architecture générale d'un agent (Fig. 8) est composée de cinq modules gérant les connaissances, la perception, la communication, le contrôle et le raisonnement de l'agent.

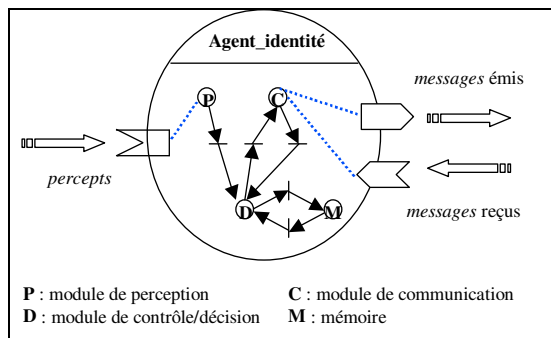


FIG. 8 – Architecture modulaire d'un agent

Nous ajouterons à cette description que les agents sont des entités hétérogènes aux modes d'interactions variés et aux comportements complexes. Une modélisation de SMA doit aussi définir le type d'organisation des agents et sa capacité d'évolution.

4.3 Des Agents Communicants et Coopérants

La communication est le principal mécanisme

d'interactions d'un agent avec la communauté des agents – en conformité avec la relation triangulaire (*Communication*, *Coopération*, *Compétition*). Pour communiquer entre eux (information ou dialogue pour la coopération), les agents expriment leurs intentions selon le langage KQML, dérivé de la théorie des actes de langages. La forme générale d'un acte de langage est donnée par J. Searle sous l'expression $F(p)$, avec $F = \{Affirmer, Demander, Promettre, Exprimer, Déclarer\}$ et p une proposition. Le format que nous avons retenu est défini par le quintuplet <intention, émetteur, récepteur, langage, message>. Il permet de représenter le contexte, l'intention et le message de la communication. Le schéma canonique de communication proposé par Abraham Moles, augmenté du feed-back et de schémas de croyances, contient les éléments intervenant dans l'acte de communication : intentionnalité et communication réciproque.

Les systèmes que nous étudions sont constitués de composants distribués, hétérogènes et autonomes. Les systèmes développés en intelligence artificielle distribuée (IAD), et notamment les SMA sont donc bien adaptés. Les comportements individuels et coopératifs des agents sont variés : planification des actions, émission et réception de documents et de messages, recherche d'information, partage de ressources, supervision de procédures. Chacun de ces services correspond à la mise en œuvre de compétences.

4.4 Le Processus d'Agentification

Les agents sont des entités possédant un certain nombre de compétences qui leur permettent de jouer des rôles dans une organisation. Ils sont regroupés au sein d'un SMA organisé hiérarchiquement. Pour la spécification du SMA nous reprenons les propositions faites dans la définition du langage A_UML [12], ainsi que nos propres propositions [6] (Fig. 9) :

- 1) réaliser le diagramme de cas d'utilisation (services rendus par le système), et pour chacun des usages identifiés réalisés les 3 phases suivantes ;
- 2) réaliser le diagramme de classes mettant en relation les agents concernés par l'usage (on peut aussi faire usage du diagramme de collaboration) ;
- 3) définir le comportement de chaque agent au moyen d'un diagramme d'états ou d'activités ;
- 4) sur la base de scénarios d'usage, réaliser les diagrammes de séquence qui précisent les échanges de messages (et leur ordonnancement) entre les agents concernés par les scénarios.

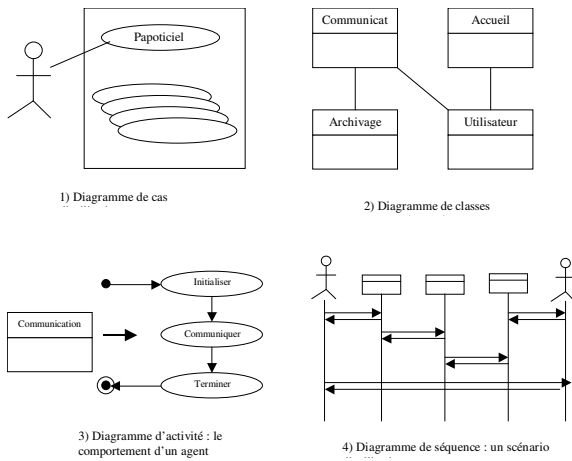


FIG. 9 – Schéma de la méthodologie proposée.

5 APPROCHE ORIENTEE AGENTS

Après avoir travaillé sur la conception objet de logiciels selon la méthode NIAM³ (Nijssen Information Analysis Method) [8], nous avons construit une méthodologie de conception agent à partir des textes de spécifications. Le traitement linguistique proposé, à l'exception de quelques nuances sur la sémantique, est relativement conforme à celui exposé précédemment (cf. §3.2). Il est entièrement orienté objets et réalisé en JAVA. Nous illustrerons la démarche avec l'exemple de la spécification présentée dans la figure 2.

5.1 Analyse Linguistique

Comme le représente la figure 10, en UML, chaque phrase d'une spécification est constituée d'une liste de mots appartenant à une catégorie grammaticale. Le fonctionnement de l'analyseur syntaxico-sémantique est analogue à celui que nous avons développé précédemment, basé sur des grammaires formelles. Ici les règles de grammaire correspondent à des schémas de phrases. Par contre, l'interprétation sémantique se distingue de la présente chaîne de traitement formel, dans la mesure où nous n'utilisons pas encore les grammaires LFG, mais les grammaires de cas, et que la représentation obtenue (Fig. 11) nous permet de traduire directement une phrase sous forme d'objets, d'agents et de relation, sans passer par les GC comme modèle de représentation intermédiaire.

Adapté à notre approche agent et à sa conception associée en AUML, l'analyseur distingue effectivement les entités correspondant à des agents ou à des objets.

³Le principe de cette méthode consiste à exprimer les spécifications à l'aide de phrases simples (*ie*, indécomposables sans perte d'information).

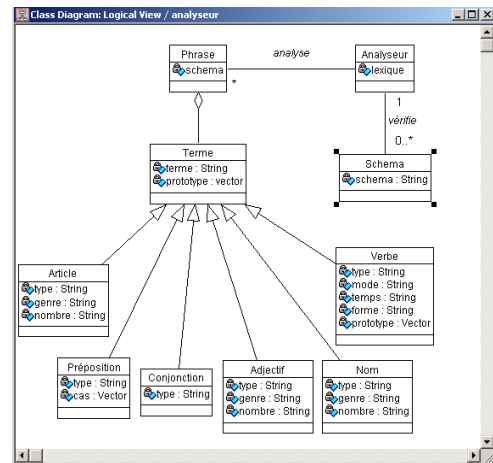


FIG. 10 – Diagramme de classes pour l'analyse syntaxico- sémantique des phrases de spécification.

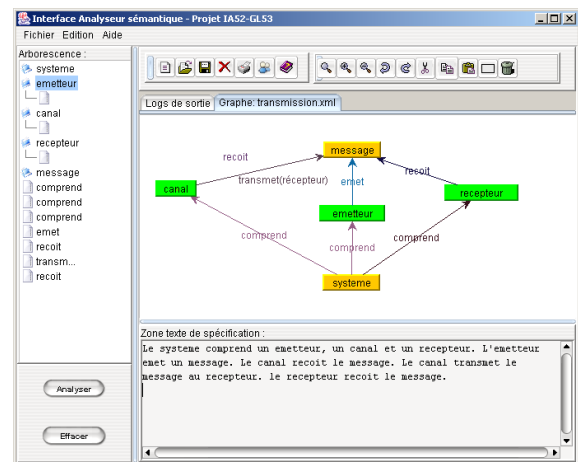


FIG. 11 – Représentation graphique de l'analyse linguistique de la spécification « Transmission ».

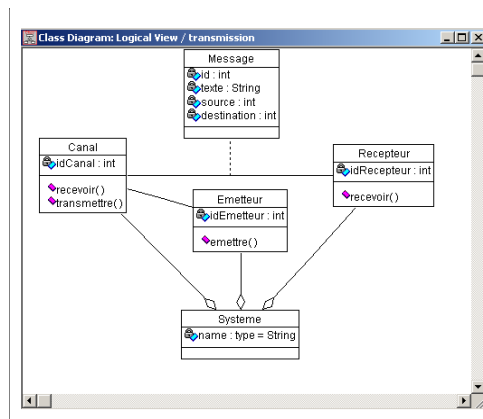


FIG. 12 – Diagramme de classes issu de l'analyse linguistique de la spécification « Transmission ».

5.2 Représentation en UML

L'analyse précédente a permis d'identifier les

éléments nécessaires à l'édition du diagramme de classes UML correspondant à un texte de spécification (classes, attributs, certaines méthodes). Notre outil ne nous permet pas encore d'éditer directement le diagramme de classes dérivé (Fig. 12) ; il s'agit d'un travail en cours. Le lecteur pourra néanmoins vérifier que ce diagramme se déduit directement des données linguistiques fournies par l'analyseur.

6 CONCLUSION

Cette communication a privilégié une présentation globale du processus de spécification à partir de textes informels rédigés en langage naturel. Les perspectives de mise en oeuvre de la première démarche de formalisation nous ont conduit à établir quatre hypothèses préalables. On retiendra plus particulièrement la deuxième hypothèse, celle qui postule l'utilisation d'une représentation intermédiaire pour passer de la spécification informelle à sa description formelle. Nous avons ainsi mis en évidence quatre niveaux de représentation d'une spécification : le texte source de la spécification écrite en langage naturel, la représentation contenant les structures linguistiques des phrases du texte, la représentation sémantique sous forme de graphes conceptuels et la représentation finale dans un modèle de description formelle tel que le langage Z. À partir de la première représentation en langage naturel, nous pouvons définir un processus dont les phases successives de traitement construisent les trois autres représentations de la spécification (processus de formalisation).

Dans le cadre de travaux plus récents nous avons partiellement réutilisé ce processus pour produire une modélisation semi-formelle en UML à partir de document de cahier des charges rédigés en langage courant. Le processus n'est pas encore achevé pour cette deuxième approche, puisque nous ne traduisons pas automatiquement la représentation syntaxico-sémantique de la spécification en diagramme UML, mais ce travail est en cours de réalisation. S'en suivra un travail sur les interactions entre les différents agents, qui nous permettra de produire, dans un premier temps, le diagramme de collaboration UML associé.

Un système, conçu selon les critères précédents, peut se présenter comme une composante intelligente d'un atelier de développement logiciel. Il en constitue le module d'élaboration des spécifications, en assistant de façon interactive un rédacteur de spécifications. Idéalement, l'entrée du système serait constituée par une spécification informelle, document rédigé en langue naturelle, et le système fournirait en sortie une spécification formelle et/ou orientée agent, après coopération avec le rédacteur. Avant de satisfaire à cette perspective, il reste néanmoins de nombreuses difficultés à résoudre c'est pourquoi, il n'est pas question pour nous d'envisager le "spécifieur

automatique" [1], dans lequel on introduirait le titre d'une spécification et qui produirait des schémas formels dépourvus d'ambiguïtés. L'outil doit donc rester au service de l'intelligence et de la créativité.

REFERENCES

- [1] R. Balzer, "A 15 year Perspective on Automatic Programming", *IEEE Transactions on Software Engineering*, SE-11(11), pp. 1257-1268, 1985.
- [2] J. Ferber, "Les systèmes multi-agents : un aperçu général", *Technique et Science Informatiques*, 16(8), 979-1012, 1997.
- [3] C. J. Fillmore, *The case for case*, Universals in Linguistic Theory, E. Bach & R. Harms Ed., New York, 1968.
- [4] A.-J. Fougères et P. Trigano, "Construction de spécifications formelles à partir des spécifications rédigées en langage naturel", *Document numérique*, Ed. Hermès, 3(3-4), p. 215-239, 1999.
- [5] A.-J. Fougères, "Model of cognitive agents to simulate complex information systems", *IEEE International Conference on Systems, Man and Cybernetics*, (SMC'02), Hammamet, Tunisia, October 6-9 2002.
- [6] A.-J. Fougères, *Des agents communicants pour simuler et détecter des épidémies*, Revue ISI, (8)1, Hermès, 2003.
- [7] M. Gross & M. Silberstein, "Outils de traitement linguistique, applications à l'analyse documentaire", *Ecole d'été Traitement des Langues Naturelles*, Trégastel, 1995.
- [8] H. Habrias, *Introduction à la spécification*, Masson, 1993.
- [9] P. Hernert, Un système d'acquisition de définitions basé sur le modèle des graphes conceptuels, Thèse de l'Université Paris XIII - Institut Gallilée, 1993.
- [10] R. Kaplan, J. Bresnan, *Lexical-Functional Grammar: a Formal System for Grammar Representation*, *The mental Representation of Grammatical Relations*, J. Bresnan Ed., MIT press, pp. 173-281, 1982.
- [11] Allen Newell, "The Knowledge Level", *Artificial Intelligence*, 18, pp. 87-127, 1982.
- [12] J. Odell, H.V.D. Parunak, B. Bauer B, *Extending UML for agents*, *Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence*, Austin, Texas, July, 30, 2000.
- [13] R.F. Paige, *Formal Method Integration via Heterogeneous Notations*, PhD dissertation, University of Toronto, 1997.
- [14] J. F. Sowa, *Conceptual Structures : Information Processing in Mind and Machine*, Addison-Wesley Publishing Company, Reading, MA, 1984.
- [15] *The SPECS Consortium, Spécification and Programming Environnement for Communication Software*, Edited by R. Reed, W. Bouma, J. Evans, M. Dauphin & M. Michel, Amsterdam, North-Holland, 1993.
- [16] J. M. Spivey, *The Z Notation*, A reference Manual, Prentice Hall, 1992.